TITLE

Data Card and Authentication Process Therefor

BACKGROUND

The present invention was conceived in the context of aircraft pilot identification, but it can be used in any situation that requires positive verification of the identity of an individual carrying an identification card.

As is well known, most identification methods that require verification can be defeated quite easily. All that is required is that a person hack into a database containing the information used for verification and insert his or her own photograph, signature, or whatever is used for verification. At an even more basic level, there are web sites that allow the user to create a counterfeit driver's license for just about any state by inserting the user's photograph, signature, and vital statistics into a form on the web site and then printing out and laminating the resulting "license".

A successful verification system should be fast and easy to use, otherwise it would constitute a bottleneck in processing large numbers of people. Ideally, the process would be set up to be started by swiping a card having a magnetic strip, barcode, optical storage area, or any combination thereof on it through a card reader such as is done today with credit cards. The process should also have more than one component of authentication, since the probability of someone hacking into more than one database increases geometrically rather than arithmetically with the increase in number of databases that need to be penetrated. The process of enrolling people in the system should also be convenient and fast in order to encourage its use. Ideally, it would also use existing equipment as much as possible in order to reduce the cost as much as possible.

OBJECTS OF THE INVENTION

Accordingly, it is an object of the present invention to provide an identification system that provides a level of security that is greater than that in present systems.

It is a further object of the present invention to provide such a system that allows quick and easy enrollment of persons in it.

It is a further object of the present invention to provide such a system that is quick and easy to use.

It is a further object of the present invention to provide such a system that uses existing computer-related equipment to a large extent.

It is a further object of the present invention to provide such a system having a verification method comprising two or more components that are stored in physically separate locations for additional security.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 shows the overall layout of the present invention.

Figure 2 shows an identification card according to the present invention.

SUMMARY

Briefly, the present invention comprises a system for creating an identification card incorporating a secure means of verifying both the card and the person presenting the card. When a person is to be enrolled in the system he or she provides unique information such as a photograph of anything desired which is to be incorporated on the card, and his or her signature, which is also unique to that person and incorporated on the card. The photograph and signature are digitally scanned and the grayscale or color plane values of certain pixels, chosen by means of a characteristic value function algorithm, are recorded on a magnetic strip, barcode, optical storage area, of a combination of these data storage media on the card along with the cardholder's name and any other desired information. The digital photograph and digital signature are recorded in a remotely located secure database. When the card is presented for authentication the holder's name is sent to the remote database, along with the

pixel values that were recorded on the card. The pixel values and identifying information are then sent, together with the stored digital photograph and digital signature, to a remote, network-inaccessible processor. The characteristic value function algorithm that was used to determine the pixel values that are stored on the card is stored at this location; the digital processor uses it to determine the pixel values from the digital photograph and signature. The processor then compares the pixel values it received with the pixel values it determined from the digital photograph and signature. If they are not identical, a message is sent back to the point of authentication request indicating that the card is not authentic. If they are identical, a message is sent back confirming the authenticity of the card and holder; the stored digital photograph and signature are also sent back and displayed, to allow further visual authentication.

DESCRIPTION OF THE PREFERRED EMBODIMENT

As shown in Figure 1 the present invention comprises a system for creating and authenticating a secure identification card. The system comprises card 10 having a magnetic strip, barcode, optical storage area, or a combination of these data storage media on it, conventional card reader 12 for reading the data on card 10, database 14 at a first remote location, remote network-inaccessible processor 16 at a second remote location, and display means 18 located near card reader 12. Card reader 12 and display means 18 are placed in locations such as controlled access areas, stores, etc. where identification cards are presented for verification. All of the components except processor 16 are connected by means 20 such as conventional telephone wires, a wireless network, or the internet. Processor 16 is connected to database 14 by secure communication link 21, as is well known in the art, so that in use processor 16 can be accessed only from the first remote location housing database 14. Isolating processor 16 in this manner assures a high level of security for the overall system. For added security, the output from processor 16 can be sent to display means 18 by a secure communication link if desired.

Figure 2 shows identification card 10 according to the present invention. It has on it certain unique information that in this example comprises photograph 22 and signature 24. Card 10 also has on it magnetic strip, barcode, optical storage area, or combination of these data storage media 26, which has encoded thereon the pixel values determined using the characteristic value function algorithm when the card was created. Card 10 may also contain any other information desired, either on its face or encoded onto storage medium 26. Photograph 22 may be of any subject desired by the owner of card 10; signature 24 is that of the card owner. Storage medium 26 also contains the information needed to begin the verification procedure.

To determine the pixel values to be encoded onto storage medium 26, photograph 22 and signature 24 are scanned to produce digital copies (not shown) which are comprised of discrete pixels, as is well known in the art. Then the digital photograph and digital signature are processed using a characteristic value function algorithm that selects certain pixels and reads their grayscale or color plane values, which are encoded as is well known in the art onto magnetic strip 26. The characteristic value function algorithm used to select the pixels may be the same for all cards or it may be varied from card to card. The characteristic value function algorithm is then stored in the same location as network-inaccessible processor 16. See the Appendix for a further explanation of the authentication process.

The digital copy of photograph 22 and digital copy of signature 24 are then sent to remote database 14 where they are stored and indexed in a way that allows them to be retrieved when desired to authenticate that particular card.

In operation, when the cardholder presents card 10 for verification it is swiped in conventional card reader 12, which then begins the verification process. Remote secure database 14 is contacted and the digital copies of photograph 22 and signature 24 are retrieved and sent to network-inaccessible processor 16. The cardholder's name and pixel values encoded on storage medium 26 are also sent to processor 16. Processor 16 applies the characteristic value function algorithm to the digital copies of photograph 22 and

signature 24 and the values of the pixels determined by the characteristic value function algorithm are read. Since a digital image is stored as a series of discrete pixel value entries in a table, the characteristic value function algorithm will determine the same pixels, and hence the same pixel values, each time; i.e., its repeatability is 100%. Thus every time card 10 is read the pixel values determined by processor 16 will be the same as those that were encoded on storage medium 26 when card 10 was created.

Processor 16 next compares the pixel values it received with the request for authentication to those it determined by applying the characteristic value function algorithm to the digital photograph and signature it received from the remote database. If they are not the same, the card is rejected as counterfeit and a message is returned to display means 18 indicating the rejection. If they are the same, the digital photograph and signature are sent back to display means 18 along with an indication that card 10 and its holder have been authenticated. Displaying photograph 22 and signature 24 on display means 18 allows further visual authentication of the card presenter.

The comparison between the pixel values determined by processor 16 and the pixel values encoded on magnetic strip 26 has been described as analytical, taking place remotely from where the card is presented. In addition, card 10 is created at another remote location, both of which insure that end-to-end security is maintained and the characteristic value function algorithm remains secret.

Also, the digital signature could be stored at a separate location to provide additional security. For even greater security the pixel values on the digital signature could be determined by a second characteristic value function algorithm which would require a second processor, stored in yet another location. Counterfeiting this latter embodiment of card 10 would require that two databases and two network-inaccessible processors be hacked into and/or that two characteristic value function algorithms, even if stored on magnetic strip 26 in assembly language, be reverse engineered, a situation that would provide a very high degree of security.

Obviously also the card could have encoded on magnetic strip 26 one or more pieces of unique information in addition to the picture and signature, thereby increasing the level of security even more.

In the following Appendix Section 1 defines the general terms used in the calculations and describes the context of the calculations. Section 2 contains a high-level overview of the process of creating the data that will be encoded on the card. Section 3 contains a short description of what is actually stored on the card. Section 4 gives a short description of the data that will be used to verify a card when it is presented for verification. Section 5 contains the core mathematics used in implementing the system. Section 6 expands on the contents of Section 5 and describes the preferred embodiment of the analytical methods behind the system of the present invention. Section 7 describes a method of preventing identical data from being encoded onto two or more cards. Section 8 describes methods of implementing the above analytical methods on a computer. Section 9 discusses the memory and storage requirements for a system as shown herein.

APPENDIX

Data Card Authentication Process

1 Introduction

Definition 1 *A* graphical value *is an integer in the range* 0 *to* 255.

Definition 2 *A* rational graphical value *is a rational number between* 0 *and* 255*, inclusive, that may be expressed as the ratio of two graphical values.*

Definition 3 *A(n) (8-bit)* chromatic value *is a real number of the form* $\frac{u}{255}$*, where* u *is a graphical value.*

Suppose a 2×2 inch image is 24-bit color scanned at ξ DPI. This means there are $4\xi^2$ pixel positions covering the entire image, each with a 24-bit color vector associated with it. Let

$$2^{q-1} < 4\xi^2 \leq 2^q$$

for some positive integer q, and let

$$(C_{i1}, C_{i2}, C_{i3})$$

be the RGB vector of (8-bit) chromatic values at pixel position i. The pixels are ordered by row/column orientation, so that pixel position (ρ, κ) would correspond to position $i = 2(\rho - 1)\xi + \kappa$. Let Ω be the totality of all (C_{i1}, C_{i2}, C_{i3}) for all pixel positions.

Let $\mathbf{I}_j^{(n)} = (i_{j1}, i_{j2}, \cdots, i_{jn})$ be a vector of choices of n-many pixel positions1 through $4\xi^2$, with repeats allowed. There are $\binom{4\xi^2}{n}$-many such choices for $I_j^{(n)}$. Let

$$\mathbf{C}_j^{(n)} = \begin{pmatrix} C_{i_{j1}1} & C_{i_{j1}2} & C_{i_{j1}3} \\ C_{i_{j2}1} & C_{i_{j2}2} & C_{i_{j2}3} \\ \vdots & \vdots & \vdots \\ C_{i_{jn}1} & C_{i_{jn}2} & C_{i_{jn}3} \end{pmatrix}_{n \times 3}$$

be the matrix of chromatic values chosen by pixel position indexed by $\mathbf{I}_j^{(n)}$. The set $\mathbf{I}_j^{(n)}$ is called the *index set* for $\mathbf{C}_j^{(n)}$.

Finally, let $\mathbf{Y}_j^{(n)} = (y_{j1}, y_{j2}, \cdots, y_{jn})^T$ be an arbitrary vector of chromatic values, called the *skew data*.

2 Calculation Methods

Define

$$b_{\mathbf{I}_j^{(n)}} = \left(\mathbf{C}_j^{(n)T}\mathbf{C}_j^{(n)}\right)^{-1}\mathbf{C}_j^{(n)T}\mathbf{Y}_j^{(n)}$$

where $b_{\mathbf{I}_j^{(n)}}$ may be viewed as a 3×1 vector or matrix.

The three values of $b_{\mathbf{I}_j^{(n)}}$ collectively represent the least-squares regression coefficients for the observed (scanned) chromatic data fitted against the skew data at positions indexed by $\mathbf{I}_j^{(n)}$. The strategic choice of the skew data $\mathbf{Y}_j^{(n)}$ for a particular $\mathbf{I}_j^{(n)}$ will ensure that identical values for $\mathbf{C}_j^{(n)}$ for a different $\mathbf{I}_j^{(n)}$ will not result in identical values for $b_{\mathbf{I}_j^{(n)}}$. In other words,

$$\mathbf{Y}_1^{(n)} \neq \mathbf{Y}_2^{(n)} \Longrightarrow b_{\mathbf{I}_1^{(n)}} \neq b_{\mathbf{I}_2^{(n)}} \text{ even if } \mathbf{C}_1^{(n)} = \mathbf{C}_2^{(n)}$$

Now let

$$\mathbb{I} = \left(\mathbf{I}_1^{(n_1)}, \mathbf{I}_2^{(n_2)} \cdots, \mathbf{I}_k^{(n_k)},\ k \geq k_0 > 1,\ k \sum_{j=1}^{k} n_j \leq 4\xi^2 \right)$$

be the set of index sets for k-many pixel choices each of size n_j. The choice for k must be at least k_0, which must be greater than 1. The value for k and for the subsidiary choices for n_j need not be globally constant; they may vary from image to image. The condition $k \sum_{j=1}^{k} n_j \leq 4\xi^2$ ensures that there are "enough pixels to go around" even if there are no repeats within the individual $\mathbf{I}_j^{(n_j)}$.

Similarly, let

$$\mathbb{C} = \left(\mathbf{C}_1^{(n_1)}, \mathbf{C}_2^{(n_2)} \cdots, \mathbf{C}_k^{(n_k)} \right)$$

with all expressions corresponding to their meanings in $\mathbb{I}$, and

$$\mathbb{Y} = \left(\mathbf{Y}_1^{(n_1)}, \mathbf{Y}_2^{(n_2)} \cdots, \mathbf{Y}_k^{(n_k)} \right)$$

3 Written and Stored Elements

The $3 \times \sum_{j=1}^{k} n_j$ matrix

$$\mathbb{B}_w = \left(b_{\mathbf{I}_1^{(n_1)}}, b_{\mathbf{I}_2^{(n_2)}}, \cdots, b_{\mathbf{I}_k^{(n_k)}} \right)$$

would be written digitally on the card, while the data

$$(\mathbb{I}, \mathbb{Y}, \Omega)$$

would be stored for further processing purposes.

4 Authentication Process

When authentication is requested, a set of values, $\mathbb{B}_r$, will be read from the card. These values will be reformatted as necessary for comparison purposes only.

The authentication process will use the stored values for $(\mathbb{I}, \Omega)$ to construct a candidate $\mathbb{C}_c$, and then $(\mathbb{C}_c, \mathbb{Y})$ will be used to calculate a candidate $\mathbb{B}_c$. If and only if

$$\mathbb{B}_r = \mathbb{B}_c$$

exactly, then the card is reported to be authenticated.

5 Calculation Issues

The exact comparison of $\mathbb{B}_r = \mathbb{B}_c$ in the authentication process leads immediately to issues of error propagation during the calculation of the individual $b_{\mathbf{I}_j^{(n)}}$. Since $\mathbf{C}_j^{(n)T}\mathbf{C}_j^{(n)}$ is a 3×3 matrix, its inverse is relatively easy to calculate with no loss of precision. The number of multiplications is limited enough to warrant an exact, rational evaluation. However, one division is needed to find $\left(\mathbf{C}_j^{(n)T}\mathbf{C}_j^{(n)}\right)^{-1}$, and this may be accomplished numerically through further rational multiplications, with limits on the introduced error.

An alternate method for performing required calculations that eliminates all error issues when comparing $\mathbb{B}_r = \mathbb{B}_c$ is to restrict the result of intermediate multiplications to chromatic values. In other words, when chromatic value $\frac{u}{255}$ is multiplied by chromatic value $\frac{v}{255}$, instead of using the value

$$\frac{uv}{(255)^2}$$

in subsequent calculations, the closest chromatic value $\frac{w}{255}$ should be used.

Definition 4 *The* chromaticon *of a real number is its closest chromatic value.*

By policy, the chromaticon for real number that are exactly half way between two chromatic values will be the **lesser** chromatic value.

Only chromaticons will be used in the calculation of the $b_{\mathbf{I}_j^{(n)}}$, whether for initial writing purposes, or for authentication.

Let

$$uv = 255k + q$$

where $0 \leq q \leq 254$, and $k = \left\lfloor \frac{uv}{255} \right\rfloor$, and $q = uv - 255k$. Since u and v are graphical values, then $0 \leq k \leq 255$. Note how there are no "half-way" issues in these circumstances. Then

$$\frac{uv}{255} = k + \frac{q}{255}$$

If $0 \leq q \leq 127$, then k is the closest graphical value to $\frac{uv}{255}$, which means $\frac{k}{255}$ is the closest chromatic value to $\frac{uv}{(255)^2}$.

If $128 \leq q \leq 254$, then $k+1$ is the closest graphical value to $\frac{uv}{255}$, which means $\frac{k+1}{255}$ is the closest chromatic value to $\frac{uv}{(255)^2}$.

These facts prove the following lemma.

Lemma 5 *The chromaticon of* $\frac{uv}{(255)^2}$ *is* $\chi\left(\frac{uv}{(255)^2}\right) = \frac{w}{255}$, *where*

$$w = \begin{cases} k, & 0 \le q \le 127 \\ k+1, & 128 \le q \le 254 \end{cases},$$

$k = \left\lfloor \frac{uv}{255} \right\rfloor$, *and* $q = uv - 255k$.

Therefore, the key to calculating $b_{\mathbf{I}_j^{(n)}}$ may be found in finding k, which involves dividing uv by 255. However, division by 255 is not as easy in digital logic as division by $256 = 2^8$. An easier way to find w is to calculate

$$w_* = \left\lfloor \frac{\left\lfloor \frac{uv+128}{256} \right\rfloor + uv + 128}{256} \right\rfloor \tag{1}$$

Calculation of w_* involves one addition of a power of 2 to uv, one 8-bit right shift, one truncation, followed by another addition of a power of 2 with uv, one more 8-bit right shift, and a final truncation, all of which may be accomplished without error very quickly in digital logic. The utility of calculating with w_* instead of w is found in the following theorem.

Theorem 6 $w_* = w$

Proof. Consider

$$uv = 255k + q$$

with all definitions as in the Lemma. Then

$$\begin{aligned} uv + 128 &= 255k + q + 128 \\ &= 256k + (q - k + 128) \end{aligned}$$

which means

$$\begin{aligned} \frac{uv+128}{256} &= k + \frac{q-k+128}{256} \\ &= k + \left(\frac{q-k}{256} + \frac{1}{2}\right) \end{aligned}$$

Since $0 \le q \le 254$ and $0 \le k \le 255$, the conceivable range of $q - k$ is -255 through 254.

Case 1: $-255 \le q - k < -128$

In this case, we have

$$-1 < -\frac{255}{256} \le \frac{q-k}{256} < -\frac{1}{2}$$

which means

$$-\frac{1}{2} < \frac{q-k}{256} + \frac{1}{2} < 0$$

Hence,

$$\left\lfloor \frac{uv+128}{256} \right\rfloor = \left\lfloor k + \left(\frac{q-k}{256} + \frac{1}{2}\right) \right\rfloor = k - 1$$

Case 2: $-128 \leq q - k < 128$

Here,

$$-\frac{1}{2} \leq \frac{q-k}{256} < \frac{1}{2}$$

which means

$$0 \leq \frac{q-k}{256} + \frac{1}{2} < 1$$

Hence,

$$\left\lfloor \frac{uv+128}{256} \right\rfloor = k$$

Case 3: $128 \leq q - k \leq 254$

Finally,

$$\frac{1}{2} \leq \frac{q-k}{256} \leq \frac{254}{256} < 1$$

which means

$$1 \leq \frac{q-k}{256} + \frac{1}{2} < 1 + \frac{1}{2}$$

Hence,

$$\left\lfloor \frac{uv+128}{256} \right\rfloor = k+1$$

To review,

$$t = \left\lfloor \frac{uv+128}{256} \right\rfloor = \begin{cases} k-1, & -255 \leq q-k < -128 \\ k, & -128 \leq q-k < 128 \\ k+1, & 128 \leq q-k \leq 254 \end{cases}$$

So,

$$t + uv + 128 = \begin{cases} 256k+q+127, & -255 \leq q-k < -128 \\ 256k+q+128, & -128 \leq q-k < 128 \\ 256k+q+129, & 128 \leq q-k \leq 254 \end{cases}$$

Recall that $0 \leq k \leq 255$ and $0 \leq q \leq 254$. Then for the first case, $-255 \leq q - k < -128$ means $k - 255 \leq q < k - 128$ so that $0 \leq q < 127$. Hence, for the first case, $0 \leq q < 127$ means $127 \leq q + 127 < 254$, so that

$$\frac{1}{2} < \frac{127}{256} \leq \frac{q+127}{256} < \frac{254}{256} < 1$$

which means

$$w_* = \left\lfloor \frac{t+uv+128}{256} \right\rfloor = \left\lfloor k + \frac{q+127}{256} \right\rfloor = k$$

for the first case, i.e., $0 \leq q \leq 127$.

Similarly, for the second case, $-128 \leq q - k < 128$ means $k - 128 \leq q < k + 128$ or $0 \leq q \leq 254$. When $0 \leq q \leq 127$, then $128 \leq q + 128 \leq 255$, which means

$$\frac{1}{2} \leq \frac{q+128}{256} \leq \frac{255}{256} < 1$$

and when $128 \leq q \leq 254$, then $256 \leq q + 128 \leq 382$, which means

$$1 \leq \frac{q+128}{256} \leq \frac{382}{256} < 1 + \frac{1}{2}$$

Hence,

$$w_* = \left\lfloor \frac{t+uv+128}{256} \right\rfloor = \left\lfloor k + \frac{q+128}{256} \right\rfloor = \begin{cases} k, & 0 \leq q \leq 127 \\ k+1, & 128 \leq q \leq 254 \end{cases}$$

for the second case.

Also, for the third case, $128 \leq q - k \leq 254$ means $k + 128 \leq q \leq k + 254$ or $128 \leq q \leq 254$. This means $257 \leq q + 129 \leq 383$, or

$$1 < \frac{257}{256} \leq \frac{q+129}{256} \leq \frac{383}{256} < 1 + \frac{1}{2}$$

Hence,

$$w_* = \left\lfloor \frac{t+uv+128}{256} \right\rfloor = \left\lfloor k + \frac{q+129}{256} \right\rfloor = k+1$$

for the third case, i.e., $128 \leq q \leq 254$.

In summary,

$$\begin{aligned} w_* &= \begin{cases} k, & 0 \leq q \leq 127 \\ k+1, & 128 \leq q \leq 254 \end{cases} \\ &= w \end{aligned}$$

■

6 Algorithms

Since all values in $\mathbb{C}$ will be chromatic values, then for purposes of calculation, only the numerator of all values need be used. In particular, since $\mathbf{C}_j^{(n)T}\mathbf{C}_j^{(n)}$ is symmetric, and the algorithm for calculating $\left(\mathbf{C}_j^{(n)T}\mathbf{C}_j^{(n)}\right)^{-1}$ involves the determinate and adjoint elements, only the upper triangular elements of $\mathbf{C}_j^{(n)T}\mathbf{C}_j^{(n)}$, using the numerator of each chromatic value, need be evaluated before the inverse is calculated.

In particular, the exact calculation of the determinate and adjoint involves three types of algebraic manipulations: Addition/Subtraction, Multiplication, and Division.

Conjecture 7 *The sum or difference of two chromatic values is a chromatic value, with the understanding that negative chromatic values and those exceeding 1 are re-expressed modulo 1.*

Proof. If $\frac{u}{255}$ and $\frac{v}{255}$ are chromatic values, then $\frac{u}{255} \pm \frac{v}{255} = \frac{u \pm v}{255}$. If $u + v > 255$, then $\frac{u}{255} + \frac{v}{255} = \frac{u+v-255}{255}$. If $u - v < 0$, then $\frac{u}{255} - \frac{v}{255} = \frac{u-v+255}{255}$. In all cases, $\frac{u}{255} \pm \frac{v}{255}$ is a chromatic value. ■

Conjecture 8 *The product of two chromatic values is a chromatic value, with the understanding that the chromaticon is the result of the multiplication.*

Conjecture 9 *The quotient of two chromatic values is a rational graphical value.*

Proof. Let $\frac{u}{255}$ and $\frac{v}{255} \neq 0$ be chromatic values. Then

$$\frac{\frac{u}{255}}{\frac{v}{255}} = \frac{u}{v}$$

This means the quotient is the ratio of two graphical values. Since $0 \leq u \leq 255$ and $1 \leq v \leq 255$, then the smallest $\frac{u}{v}$ may become is $0 = \frac{0}{v}$, and the largest it may become is $255 = \frac{255}{1}$. Hence, the quotient is a rational graphical value. ■

Notation 10 *If ζ_1 and ζ_2 are chromatic values, then $\chi(\zeta_1\zeta_2) = \chi\zeta_1\zeta_2$ is the chromaticon of ζ_1 and ζ_2.*

Lemma 11 $\left|\mathbf{C}_j^{(n)T}\mathbf{C}_j^{(n)}\right|$ *is a chromatic value.*

Proof. We have

$$\mathbf{C}_j^{(n)T}\mathbf{C}_j^{(n)} = \begin{pmatrix} \sum_{k=1}^{n} \chi C_{i_{jk}1}^2 & \sum_{k=1}^{n} \chi C_{i_{jk}1}C_{i_{jk}2} & \sum_{k=1}^{n} \chi C_{i_{jk}1}C_{i_{jk}3} \\ \sum_{k=1}^{n} \chi C_{i_{jk}2}C_{i_{jk}1} & \sum_{k=1}^{n} \chi C_{i_{jk}2}^2 & \sum_{k=1}^{n} \chi C_{i_{jk}2}C_{i_{jk}3} \\ \sum_{k=1}^{n} \chi C_{i_{jk}3}C_{i_{jk}1} & \sum_{k=1}^{n} \chi C_{i_{jk}3}C_{i_{jk}2} & \sum_{k=1}^{n} \chi C_{i_{jk}3}^2 \end{pmatrix} \bmod 1 \tag{2}$$

by the policies of Conjectures 7-8.

Expanding $\mathbf{C}_j^{(n)T}\mathbf{C}_j^{(n)}$ by its first column, we have

$$\begin{aligned} \left|\mathbf{C}_j^{(n)T}\mathbf{C}_j^{(n)}\right| = {} & \chi \sum_{k=1}^{n} \chi C_{i_{jk}1}^2 \begin{bmatrix} \chi \sum_{k=1}^{n} \chi C_{i_{jk}2}^2 \sum_{k=1}^{n} \chi C_{i_{jk}3}^2 \\ -\chi \left(\sum_{k=1}^{n} \chi \left(C_{i_{jk}2}C_{i_{jk}3} \right) \right)^2 \end{bmatrix} \bmod 1 \\ & -\chi \sum_{k=1}^{n} \chi C_{i_{jk}2}C_{i_{jk}1} \begin{bmatrix} \chi \sum_{k=1}^{n} \chi C_{i_{jk}1}C_{i_{jk}2} \sum_{k=1}^{n} \chi C_{i_{jk}3}^2 \\ -\chi \sum_{k=1}^{n} \chi C_{i_{jk}1}C_{i_{jk}3} \sum_{k=1}^{n} \chi C_{i_{jk}3}C_{i_{jk}2} \end{bmatrix} \bmod 1 \\ & +\chi \sum_{k=1}^{n} \chi C_{i_{jk}3}C_{i_{jk}1} \begin{bmatrix} \chi \sum_{k=1}^{n} \chi C_{i_{jk}1}C_{i_{jk}2} \sum_{k=1}^{n} \chi C_{i_{jk}2}C_{i_{jk}3} \\ -\chi \sum_{k=1}^{n} \chi C_{i_{jk}1}C_{i_{jk}3} \sum_{k=1}^{n} \chi C_{i_{jk}2}^2 \end{bmatrix} \bmod 1 \end{aligned} \tag{3}$$

By the policies of Conjectures 7-8, we see immediately that $\left|\mathbf{C}_j^{(n)T}\mathbf{C}_j^{(n)}\right|$ is a chromatic value. ■

Lemma 12 *The adjoint of* $\mathbf{C}_j^{(n)T}\mathbf{C}_j^{(n)}$ *is a* 3×3 *matrix of chromatic values.*

Proof. As before, we have

$$\mathbf{C}_j^{(n)T}\mathbf{C}_j^{(n)} = \begin{pmatrix} \sum_{k=1}^{n} \chi C_{i_{jk}1}^2 & \sum_{k=1}^{n} \chi C_{i_{jk}1}C_{i_{jk}2} & \sum_{k=1}^{n} \chi C_{i_{jk}1}C_{i_{jk}3} \\ \sum_{k=1}^{n} \chi C_{i_{jk}2}C_{i_{jk}1} & \sum_{k=1}^{n} \chi C_{i_{jk}2}^2 & \sum_{k=1}^{n} \chi C_{i_{jk}2}C_{i_{jk}3} \\ \sum_{k=1}^{n} \chi C_{i_{jk}3}C_{i_{jk}1} & \sum_{k=1}^{n} \chi C_{i_{jk}3}C_{i_{jk}2} & \sum_{k=1}^{n} \chi C_{i_{jk}3}^2 \end{pmatrix} \bmod 1$$

Let

$$\begin{aligned}
A &= \chi \sum_{k=1}^{n} \chi C_{i_{jk}2}^2 \sum_{k=1}^{n} \chi C_{i_{jk}3}^2 - \chi \left(\sum_{k=1}^{n} \chi \left(C_{i_{jk}2}C_{i_{jk}3} \right) \right)^2 \bmod 1 \\
B &= \chi \sum_{k=1}^{n} \chi C_{i_{jk}1}C_{i_{jk}3} \sum_{k=1}^{n} \chi C_{i_{jk}3}C_{i_{jk}2} - \chi \sum_{k=1}^{n} \chi C_{i_{jk}1}C_{i_{jk}2} \sum_{k=1}^{n} \chi C_{i_{jk}3}^2 \bmod 1 \\
D &= \chi \sum_{k=1}^{n} \chi C_{i_{jk}1}C_{i_{jk}2} \sum_{k=1}^{n} \chi C_{i_{jk}2}C_{i_{jk}3} - \chi \sum_{k=1}^{n} \chi C_{i_{jk}1}C_{i_{jk}3} \sum_{k=1}^{n} \chi C_{i_{jk}2}^2 \bmod 1 \\
E &= \chi \sum_{k=1}^{n} \chi C_{i_{jk}2}C_{i_{jk}3} \sum_{k=1}^{n} \chi C_{i_{jk}3}C_{i_{jk}1} - \chi \sum_{k=1}^{n} \chi C_{i_{jk}2}C_{i_{jk}1} \sum_{k=1}^{n} \chi C_{i_{jk}3}^2 \bmod 1 \\
F &= \chi \sum_{k=1}^{n} \chi C_{i_{jk}1}^2 \sum_{k=1}^{n} \chi C_{i_{jk}3}^2 - \chi \left(\sum_{k=1}^{n} \chi \left(C_{i_{jk}1}C_{i_{jk}3} \right) \right)^2 \bmod 1 \\
G &= \chi \sum_{k=1}^{n} \chi C_{i_{jk}1}C_{i_{jk}3} \sum_{k=1}^{n} \chi C_{i_{jk}2}C_{i_{jk}1} - \chi \sum_{k=1}^{n} \chi C_{i_{jk}1}^2 \sum_{k=1}^{n} \chi C_{i_{jk}2}C_{i_{jk}3} \bmod 1 \\
H &= \chi \sum_{k=1}^{n} \chi C_{i_{jk}2}C_{i_{jk}1} \sum_{k=1}^{n} \chi C_{i_{jk}3}C_{i_{jk}2} - \chi \sum_{k=1}^{n} \chi C_{i_{jk}2}^2 \sum_{k=1}^{n} \chi C_{i_{jk}3}C_{i_{jk}1} \bmod 1 \\
K &= \chi \sum_{k=1}^{n} \chi C_{i_{jk}1}C_{i_{jk}2} \sum_{k=1}^{n} \chi C_{i_{jk}3}C_{i_{jk}1} - \chi \sum_{k=1}^{n} \chi C_{i_{jk}1}^2 \sum_{k=1}^{n} \chi C_{i_{jk}3}C_{i_{jk}2} \bmod 1 \\
L &= \chi \sum_{k=1}^{n} \chi C_{i_{jk}1}^2 \sum_{k=1}^{n} \chi C_{i_{jk}2}^2 - \chi \left(\sum_{k=1}^{n} \chi \left(C_{i_{jk}1}C_{i_{jk}2} \right) \right)^2 \bmod 1
\end{aligned} \tag{4}$$

Then the adjoint of $\mathbf{C}_j^{(n)T}\mathbf{C}_j^{(n)}$ is

$$J\left(\mathbf{C}_j^{(n)T}\mathbf{C}_j^{(n)}\right) = \begin{pmatrix} A & B & D \\ E & F & G \\ H & K & L \end{pmatrix} \tag{5}$$

which is a 3×3 matrix of chromatic values, by the policies of Conjectures 7-8. ■

Theorem 13 $b_{\mathbf{I}_j^{(n)}}$ *is a* 3×1 *vector of rational graphical values.*

Proof. We have

$$b_{\mathbf{I}_j^{(n)}} = \left(\mathbf{C}_j^{(n)T}\mathbf{C}_j^{(n)}\right)^{-1} \mathbf{C}_j^{(n)T}\mathbf{Y}_j^{(n)}$$

Since

$$\left(\mathbf{C}_j^{(n)T}\mathbf{C}_j^{(n)}\right)^{-1} = \frac{1}{\left|\mathbf{C}_j^{(n)T}\mathbf{C}_j^{(n)}\right|} J\left(\mathbf{C}_j^{(n)T}\mathbf{C}_j^{(n)}\right)$$

we have

$$\left|\mathbf{C}_j^{(n)T}\mathbf{C}_j^{(n)}\right| b_{\mathbf{I}_j^{(n)}} = J\left(\mathbf{C}_j^{(n)T}\mathbf{C}_j^{(n)}\right)\mathbf{C}_j^{(n)T}\mathbf{Y}_j^{(n)}$$

Now

$$J\left(\mathbf{C}_j^{(n)T}\mathbf{C}_j^{(n)}\right)\mathbf{C}_j^{(n)T} = \begin{pmatrix} \chi AC_{i_{j1}1} & \chi AC_{i_{j2}1} & \cdots & \chi AC_{i_{jn}1} \\ +\chi BC_{i_{j1}2} & +\chi BC_{i_{j2}2} & & +\chi BC_{i_{jn}2} \\ +\chi DC_{i_{j1}3} & +\chi DC_{i_{j2}3} & & +\chi DC_{i_{jn}3} \\ \chi EC_{i_{j1}1} & \chi EC_{i_{j2}1} & & \chi EC_{i_{jn}1} \\ +\chi FC_{i_{j1}2} & +\chi FC_{i_{j2}2} & \cdots & +\chi FC_{i_{jn}2} \\ +\chi GC_{i_{j1}3} & +\chi GC_{i_{j2}3} & & +\chi GC_{i_{jn}3} \\ \chi HC_{i_{j1}1} & \chi HC_{i_{j2}1} & & \chi HC_{i_{jn}1} \\ +\chi KC_{i_{j1}2} & +\chi KC_{i_{j2}2} & \cdots & +\chi KC_{i_{jn}2} \\ +\chi LC_{i_{j1}3} & +\chi LC_{i_{j2}3} & & +\chi LC_{i_{jn}3} \end{pmatrix} \bmod 1$$

and

$$\left|\mathbf{C}_j^{(n)T}\mathbf{C}_j^{(n)}\right| b_{\mathbf{I}_j^{(n)}} = \begin{pmatrix} \chi\begin{pmatrix} \chi AC_{i_{j1}1} \\ +\chi BC_{i_{j1}2} \\ +\chi DC_{i_{j1}3} \end{pmatrix} y_1 + \chi\begin{pmatrix} \chi AC_{i_{j2}1} \\ +\chi BC_{i_{j2}2} \\ +\chi DC_{i_{j2}3} \end{pmatrix} y_2 \\ \qquad + \cdots + \chi\begin{pmatrix} \chi AC_{i_{jn}1} \\ +\chi BC_{i_{jn}2} \\ +\chi DC_{i_{jn}3} \end{pmatrix} y_n \\ \chi\begin{pmatrix} \chi EC_{i_{j1}1} \\ +\chi FC_{i_{j1}2} \\ +\chi GC_{i_{j1}3} \end{pmatrix} y_1 + \chi\begin{pmatrix} \chi EC_{i_{j2}1} \\ +\chi FC_{i_{j2}2} \\ +\chi GC_{i_{j2}3} \end{pmatrix} y_2 \\ \qquad + \cdots + \chi\begin{pmatrix} \chi EC_{i_{jn}1} \\ +\chi FC_{i_{jn}2} \\ +\chi GC_{i_{jn}3} \end{pmatrix} y_n \\ \chi\begin{pmatrix} \chi HC_{i_{j1}1} \\ +\chi KC_{i_{j1}2} \\ +\chi LC_{i_{j1}3} \end{pmatrix} y_1 + \chi\begin{pmatrix} \chi HC_{i_{j2}1} \\ +\chi KC_{i_{j2}2} \\ +\chi LC_{i_{j2}3} \end{pmatrix} y_2 \\ \qquad + \cdots + \chi\begin{pmatrix} \chi HC_{i_{jn}1} \\ +\chi KC_{i_{jn}2} \\ +\chi LC_{i_{jn}3} \end{pmatrix} y_n \end{pmatrix} \bmod 1 \tag{6}$$

which is a 3×1 vector of chromatic values by the policies of Conjectures 7-8.

Since $\left|\mathbf{C}_j^{(n)T}\mathbf{C}_j^{(n)}\right|$ is a (scalar) chromatic value, then by Conjecture 9, $b_{\mathbf{I}_j^{(n)}}$ is a 3×1 vector of rational graphical values. ■

The $3 \times \sum_{j=1}^{k} n_j$ matrix $\mathbb{B}_w = \left(b_{\mathbf{I}_1^{(n_1)}}, b_{\mathbf{I}_2^{(n_2)}}, \cdots, b_{\mathbf{I}_k^{(n_k)}}\right)$ is therefore a matrix of rational graphical values, where each $b_{\mathbf{I}_j^{(n_j)}}$ has a common graphical value denominator among its three components (see the proof of Theorem 13 on page 8).

For simplicity, the matrix $\mathbb{B}_w$ will be written as a stream of $4k$-many graphical values in the following order:

$$\mathbb{B}_w = \left\{ \beta_1, \alpha_{11}, \alpha_{12}, \alpha_{13}, \alpha_{21}, \beta_2, \alpha_{22}, \alpha_{23}, \cdots, \underbrace{\alpha_{j1}, \alpha_{j2}, \alpha_{j3}, \beta_j}_{\beta_j \text{ in position } ((j-1) \bmod 4)+1}, \cdots \right\}$$

where

$$b_{\mathbf{I}_j^{(n_j)}} = \left(\frac{\alpha_{j1}}{\beta_j}, \frac{\alpha_{j2}}{\beta_j}, \frac{\alpha_{j3}}{\beta_j} \right)$$

Since each α_{jk} and β_j is an 8-bit integer, then $\mathbb{B}_w$ will be of binary length $8 \times 4k = 32k$.

7 Duplicate Prevention

To review, the strategic choice of the skew data $\mathbf{Y}_j^{(n)}$ for a particular $\mathbf{I}_j^{(n)}$ will ensure that identical values for $\mathbf{C}_j^{(n)}$ for a different $\mathbf{I}_j^{(n)}$ will not result in identical values for $b_{\mathbf{I}_j^{(n)}}$. To be specific, suppose $\mathbb{B}_{w_1}$ has been calculated for a particular k_1-many choices $(\mathbb{Y}_1, \mathbb{I}_1)$ given $\mathbb{C}_1$. Suppose a different color matrix $\mathbb{C}_2$ has produced an identical $\mathbb{B}_{w_1}$ for another k_1-many choices $(\mathbb{Y}_2, \mathbb{I}_2)$ given $\mathbb{C}_2$. This event is **highly** unlikely to occur with a random choice of $\mathbb{Y}_2$: Approximately one chance in 2^{32k_1} for k_1-many choices of $\left(\mathbf{Y}_j^{(n_j)}, \mathbf{I}_j^{(n_j)}\right)$. However, if this event does occurs, then one more $\left(\mathbf{Y}_{k_1+1}^{(n_{k_1+1})}, \mathbf{I}_{k_1+1}^{(n_{k_1+1})}\right)$ will be chosen to calculate one more $b_{\mathbf{I}_{k_1+1}^{(n)}}$ to produce (k_1+1)-many choices $(\mathbb{Y}_2^*, \mathbb{I}_2^*)$ given $\mathbb{C}_2$. This will produce a new $\mathbb{B}_{w_2}$ that is necessarily different from $\mathbb{B}_{w_1}$, since the binary representations will necessarily be of different lengths. Note how this adjustment need only be performed once to prevent all duplicate $\mathbb{B}_w$.

8 Calculation Methods

The calculation methods found throughout this document may be implemented with digital logic operating on registers. The details found below are flexible enough to allow for higher precision color scanning, e.g., 48-bit color rather than 24-bit, or more sophisticated graphical effects, e.g., dithering.

8.1 Encoding $\mathbb{B}_w$

Given n, $\mathbf{I}_j^{(n)}$, $\mathbf{Y}_j^{(n)}$, and $\mathbf{C}_j^{(n)}$, the following steps make $\alpha_{j1}, \alpha_{j2}, \alpha_{j3}, \beta_j$ available for encoding, for $j = 1, 2, \ldots, k$. All calculations are performed according to the policies of Conjectures 7-9, i.e., for chromaticons mod 1.

1. Calculate the 3×3 matrix $\mathbf{C}_j^{(n)T}\mathbf{C}_j^{(n)}$ [see (2) on page 7].

2. Calculate the scalar $\beta_j = \left|\mathbf{C}_j^{(n)T}\mathbf{C}_j^{(n)}\right|$ [see (3) on page 7].

3. Calculate the 3×3 matrix $J\left(\mathbf{C}_j^{(n)T}\mathbf{C}_j^{(n)}\right)$ [see (5) on page 8].

4. Calculate the 3×1 vector $(\alpha_{j1}, \alpha_{j2}, \alpha_{j3})^T = \beta_j b_{\mathbf{I}_j^{(n)}}$ [see (6) on page 9].

5. Repeat Steps 1-4 for incremented j.

8.2 Chromaticon $\chi\zeta_1\zeta_2$

Given two chromatic values ζ_1 and ζ_2, the following steps make $\chi\zeta_1\zeta_2$ available for further calculation. All calculations make reference to 4-digit hexadecimal encoded (16-bit) assembly level instructions in a specially designed Application Specific Integrated Circuit (ASIC) with a Severely Reduced Instruction Set (SRIS). All underlying values in the calculations are 16-bit binary numbers [see (1) on page 4].

1. Multiply ζ_1 and ζ_2 as unsigned (16-bit) integers. Save the result as r_1.

2. Add 0000000010000000 (decimal 128) to r_1; this gives r_2.

3. Right shift r_2 by 8 places; this gives r_3.

4. Add r_1 to r_3, which gives r_4.

5. Add 0000000010000000 (decimal 128) to r_4; this gives r_5.

6. Right shift r_5 by 8 places; this gives $\chi\zeta_1\zeta_2$.

8.3 Addition Modulo 1

Given two chromatic values ζ_1 and ζ_2, the following steps make $\zeta_1 + \zeta_2$ available for further calculation. All calculations make references as in the previous subsection [see Conjecture 7 on page 6]. Note how $\zeta_1 + \zeta_2$ can never become negative, and that overflow cannot happen when adding 8-bit integers within 16-bit digital logic.

1. Add ζ_1 and ζ_2 as unsigned (16-bit) integers. Save the result as r_1.

2. If $r_1 > 255$, then subtract 0000000011111111 (decimal 255) from r_1; this gives $\zeta_1 + \zeta_2$.

Note also that underflow is prevented in Step 2 by its conditional nature.

8.4 Subtraction Modulo 1

Given two chromatic values ζ_1 and ζ_2, the following steps make $\zeta_1 - \zeta_2$ available for further calculation. All calculations make references as in the previous subsections [see Conjecture 7 on page 6]. Note how $\zeta_1 - \zeta_2$ can never become greater than 255.

1. If $\zeta_1 > \zeta_2$, subtract ζ_2 from ζ_1 as unsigned (16-bit) integers. The result is $\zeta_1 - \zeta_2$.
2. If $\zeta_1 < \zeta_2$, add 0000000011111111 (decimal 255) to ζ_1; call it r_1. Now subtract ζ_2 from r_1 as unsigned (16-bit) integers. The final result is $\zeta_1 - \zeta_2$.
3. If $\zeta_1 = \zeta_2$, then $\zeta_1 - \zeta_2 = 0$ (as an unsigned 16-bit integer).

Note also that underflow is prevented by the conditional logic of Steps 1-3.

9 Storage and Transmission Requirements

Each image contains $4\xi^2$ pixel positions, not all of which need to be stored to enable authentication under subsequent processing (even though there is no loss of processing capabilities if data were stored for all pixel positions). At minimum, since $\sum_{j=1}^{k} n_j$-many pixel positions are used in the calculation of $\mathbb{B}_w$, then $24 \sum_{j=1}^{k} n_j$ bits need to be stored to account for the color matrix $\mathbb{C}$ from the image Ω. Furthermore, $q \sum_{j=1}^{k} n_j$-many bits need to be stored to account for $\mathbb{I}$, and $8 \sum_{j=1}^{k} n_j$-many bits need to be stored to account for $\mathbb{Y}$.

Therefore, to store $(\mathbb{I}, \mathbb{Y}, \Omega)$ for authentication purposes, space for $(32 + q) \sum_{j=1}^{k} n_j$ bits are needed. However, in the case that Ω must be stored in its entirety for transmission purposes, then only

$$(8 + q) \sum_{j=1}^{k} n_j$$

bits need be stored for authentication purposes.

Under the common use of $\xi = 300$ (DPI), then $q = 19$ (since $2^{18} = 262144 < 360000 = 4\,(300)^2 \leq 524288 = 2^{19}$). If $k = 5$ sets of $n_j \equiv 8$ pixel positions are used (for a total of 40 positions), then

$$(8 + q) \sum_{j=1}^{k} n_j = 27 \times 40 = 1080$$

bits are needed to store one processing set. This is slightly more than 1 kilobyte of information per processing set.

However, $24 \times 4 \times (300)^2 = 8,640,000$ bits, or approximately 8 megabytes are needed to store and transmit the entire image for end-processing purposes. This number may be significantly reduced by compression algorithms.

Alternatively, scanning may proceed at $\xi = 72$ (DPI), which is the typical resolution for monitor display. This means $24 \times 4 \times (72)^2 = 20736$ bits are needed to store the entire image, or slightly more than 20 kilobytes per stored image. Even with only 20000 (or so) pixel positions from which to choose, at most 40 are needed under the circumstances described above, and $\binom{20000}{40}$ is astronomically larger than the expected number of individual images to be processed.